TU Delft Delft University of Technology

OPTIMISATION FOR SYSTEMS AND CONTROL

(SC42056)

Linear and Quadratic Programming Assignment

Students

Daniel Varela - 5321263

Rohan Chandrashekar - 5238382

Teacher

Bart De Schutter

October 8, 2020

Contents

1 Introduction

Maintenance of indoor ambient is a modern yet, compelling requirement as it caters to various factors, most primarily relevant to human health and productivity. This can be achieved through architectural styles that provide optimum ventilation, lighting, and most importantly, controlling the thermal comforts of a closed environment. In this report, we shall be looking at optimizing one such HVAC (Heating, Ventilation, and Air-Conditioning) system, leading to its optimum usage, given the constraints on the total number of units to be installed and budget for its installation, operation, and maintenance.

2 Task 1

2.1 Task 1(a)

2.1.1 Problem Definition

The problem definition requires us to find the optimal number of air conditioning units of types X and Y for a HVAC system such that the maximum power of both units is obtained, given the constraints on the total number of units and the budget for installation of the air conditioners.
Our optimization problem is defined as follows:

$$\max_{X,Y}(XP_X + YP_Y) \tag{1}$$

where P_X and P_Y are the max power ratings of the respective air conditioning units.

2.1.2 Constraints Definition

The problem comprises of constraints that have been defined for the total number of units that can be installed and the budget allocated for the installation of these units. These constraints are written as follows:

$$\begin{cases} X + Y \leq 12 \ \ (Maximum\ number\ of\ units\ to\ be\ installed) \\ XPr_X + YPr_Y \leq 24000 + E_1 \ \ (Max\ budget\ (B)\ of\ units\ to\ be\ installed) \end{cases} \tag{2}$$

Where Pr_X and Pr_Y are the individual unit costs of the respective air conditioning units.

2.1.3 Linear Programming

The defined problem can be formulated as a LP (Linear Programming), where the function to be maximized with respect to X and Y can be written as follows:

$$f(X,Y) = XP_X + YP_Y \tag{3}$$

$f(X,Y)$ is a linear equation, having been written as a combination of constants and linear variables. The constraints in (2) apply to the LP and since X and Y are number of air conditioning units, negative values are not applicable. Hence, the following constraints also apply:

$$\begin{cases} X, Y \geq 0 \end{cases} \Leftrightarrow \begin{cases} -X, -Y \leq 0 \end{cases} \tag{4}$$

Given the constraints also comprise of linear/affine inequalities, in combination with (3) being linear, allows us to conclude that our problem, with linear constraints can indeed be seen as a LP problem. We start by defining our optimization vector. This contains the variables we want to optimize in our problem.

$$x = [X, Y] \tag{5}$$

The maximization function needs to be brought to the standard form in order to solve the problem. This can be done by minimizing the negative of the equation to optimize. Thus, the optimization problem to solve is:

$$\min_{x} f(x) = c^{\mathsf{T}} x \Leftrightarrow \min_{X,Y} (-XP_X - YP_Y) \tag{6}$$

This allows us to write the matrix c as:

$$c^{\mathsf{T}} = \begin{bmatrix} -P_X & -P_Y \end{bmatrix} \tag{7}$$

The inequality constraints can be written to take the form $Ax \leq b$, with:

$$A = \begin{bmatrix} Pr_X & Pr_Y \\ 1 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} \qquad b = \begin{bmatrix} 24000 + 300E_1 \\ 12 \\ 0 \\ 0 \end{bmatrix} \tag{8}$$

Having defined our problem with the function to minimize with respect to c^{T} and x and our inequality matrices being A and b, one can now use them to solve the LP problem in `MATLAB` using the command `linprog`.

2.2 Task 1(b)

The above formulated matrices are used to solve the LP Problem. We compute the budget with the values for the parameters $E_1 = 5$, $E_2 = 14$ and $E_3 = 5$. Knowing $P_X = 4000$W and $P_Y = 2500$W, we can construct matrix c as stated in (7) and with the values of $Pr_X = 3000€$ and $Pr_Y = 1500€$, we can construct matrices A and b as outlined in (8). One can now solve the LP problem using the following `MATLAB` command:

```
[x1a,fval,exitflag] = linprog(cTa,Aa,ba,[],[],[]);
```
(9)

`x1a` is a vector which contains the optimized values for the units X and Y. The output from the `linprog` command is given below:

$$X = 5\ units,\ Y = 7\ units$$

$$Budget\ spent\ (B) = 25,500€$$

The maximum power obtained is then calculated which is $37.5KW$. Note that if we were to choose $X = 6$ and $Y = 6$, we would get a maximum power of $39KW$, while staying within the maximum allowed number of units. However, this violates the budget constraint of $24000 + 300E_1$ ($27,000€ > 25,500€$). Hence, the budget (B) is the limiting constraint for the installation.

Further, we should also note that, because theses values are integers, they are the optimal values for X and Y and no post-processing is needed. However, this is not necessarily always the case, and the values could very well be non-integer results, as we'll see in 2.3.

2.3 Task 1(c)

2.3.1 Problem Definition

This task requires one to take into account the changes in maintenance cost and the extra budget to decide on the optimal number of years to keep the air conditioning units durably functional and to maximize the total power of our installation. Therefore, our optimization problem, similar to the previous tasks, is the following:

$$\max_{X_i,Y_i}(X_iP_X + Y_iP_Y) \tag{10}$$

Where X_i and Y_i are the number of units of X and Y corresponding to each of the years, with $i = 1, 2, (\dots), 10.$

2.3.2 Constraints Definition

The constraints indicated are the same, qualitatively, for every year. For each of the 10 years, our total number of units cannot surpass the allowed maximum of 12 units. Further, for every year, the total price of the units, to which we add the maintenance costs for that year, and all years prior, cannot be higher than our total budget, to which we add a fixed sum every year as well.

In practise, this means that one of our constraints remains constant, but the other one varies according to the year we're in. We can write the constraints as follows:

$$\begin{cases} X_i + Y_i \leq 12 \\ X_i Pr_{Xi} + Y_i Pr_{Yi} \leq B_i \end{cases} \tag{11}$$

With:

$$B_i = (24000 + 300E_1 + \sum_{k=1}^{i} 4000 + 100E_1)€ \tag{12}$$

$$Pr_{Xi} = 3000 + \sum_{k=1}^{i} M_{Xk}\ € \tag{13}$$

$$Pr_{Yi} = 3000 + \sum_{k=1}^{i} M_{Yk}\ € \tag{14}$$

Finally, M_{Xk} and M_{Yk} represent the maintenance costs per year, are given in the problem and can be consulted below:

Year (k)	1	2	3	4	5
X_k	$200 + E_2$	$200 + 2E_2$	$200 + 3E_2$	$300 + 4E_2$	$300 + 5E_2$
Y_k	$50 + E_2$	$50 + 2E_2$	$100 + 3E_2$	$150 + 4E_2$	$150 + 5E_2$
Year (k)	6	7	8	9	10
X_k	$400 + 5E_2$	$500 + 5E_2$	$600 + 5E_2$	$700 + 5E_2$	$800 + 5E_2$
Y_k	$200 + 5E_2$	$250 + 5E_2$	$300 + 5E_2$	$350 + 5E_2$	$400 + 5E_2$

Table 1: Maintenance values X_k and Y_k for each of the 10 years.

2.3.3 Linear Programming

We start by recalling our optimization problem is the maximization of the following function with respect to X_i and Y_i:

$$f(X_i, Y_i) = X_i P_{Xi} + Y_i P_{Yi} \tag{15}$$

$f(X_i, Y_i)$ is, obviously, a linear equation. The problem, by itself, has no solution, because no finite minimum exists. However, the constraints in (11) apply. Additionally , similar to Task 1(a), the following constraints also apply:

$$\begin{cases} X_i \geq 0 \\ Y_i \geq 0 \end{cases} \Leftrightarrow \begin{cases} -X_i \leq 0 \\ -Y_i \leq 0 \end{cases} \tag{16}$$

All four constraints can are either linear or affine inequalities. This fact, in combination with (15) being linear allows us to conclude that our linear problem, with linear/affine constraints can indeed be seen as a LP Problem. As seen previously, this maximization task can be brought to standard form by minimizing the negative of the equation to optimize. Thus, the optimization problem to solve is:

$$\min_x f(x) = c^\intercal x \Leftrightarrow \min_{X_i, Y_i} (-X_i P_X - Y_i P_Y) \tag{17}$$

We start by defining our optimization vector. This will contain all the variables we want to optimize in our problem.

$$x = [X_i, Y_i] \tag{18}$$

From this form of the problem, we can write the matrix c as:

$$c^\intercal = \begin{bmatrix} -P_{Xi} & -P_{Yi} \end{bmatrix} \tag{19}$$

The inequality constraints can be re-written to take the form $Ax \leq b$, with:

$$A = \begin{bmatrix} Pr_{Xi} & Pr_{Yi} \\ 1 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} \qquad b = \begin{bmatrix} B_i \\ 12 \\ 0 \\ 0 \end{bmatrix} \tag{20}$$

Finally, our problem is fully defined. We can now use these matrices to solve 10 different LP problems using the `MATLAB` using the command `linprog`.

The same results, however, can be obtained from a single LP problem, if we notice that all of these problems have no effect on each other, and are, in effect, disjunct. The variables, year after year, have no effect on each other. We can simply solve them by joining all of our LP problems as a single LP. This in turn can be defined as:

$$\min_x f(x) = c^\intercal x \Leftrightarrow \min_{\substack{X_1,(\dots),X_{10}, \\ Y_1,(\dots),Y_{10}}} \sum_{i=1}^{10} (-X_i P_X - Y_i P_Y) \tag{21}$$

We can define our optimization vector x and our coefficients vector c, for convenience, as:

$$x = \begin{bmatrix} X_1 & Y_1 & X_2 & Y_2 & \dots & X_{10} & Y_{10} \end{bmatrix} \qquad c^{\mathsf{T}} = \begin{bmatrix} P_X & P_Y & P_X & P_Y & \dots & P_X & P_Y \end{bmatrix} \tag{22}$$

The corresponding inequality matrices take the same form as above, $Ax \leq b$, but now $A \in \mathbb{R}^{40\times 20}$ and $b \in \mathbb{R}^{20\times 1}$:

$$A = \begin{bmatrix} Pr_{X1} & Pr_{Y1} & (\dots) & 0 & 0 \\ (\dots) & (\dots) & (\dots) & (\dots) & (\dots) \\ 0 & 0 & (\dots) & Pr_{X10} & Pr_{Y10} \\ 1 & 1 & (\dots) & 0 & 0 \\ (\dots) & (\dots) & (\dots) & (\dots) & (\dots) \\ 0 & 0 & (\dots) & 1 & 1 \\ -1 & 0 & (\dots) & 0 & 0 \\ (\dots) & (\dots) & (\dots) & (\dots) & (\dots) \\ 0 & 0 & (\dots) & 0 & -1 \\ -1 & 0 & (\dots) & 0 & 0 \\ (\dots) & (\dots) & (\dots) & (\dots) & (\dots) \\ 0 & 0 & (\dots) & 0 & -1 \end{bmatrix} \qquad b = \begin{bmatrix} B_1 \\ (\dots) \\ B_{10} \\ 12 \\ (\dots) \\ 12 \\ 0 \\ (\dots) \\ 0 \\ 0 \\ (\dots) \\ 0 \end{bmatrix} \tag{23}$$

This is more compact, but a more complex definition of the problem. The formulation as a single LP does not yield the optimal (X_i, Y_i), but it allows us to obtain the same results as solving 10 separate LP's. The process through which the results are obtained, as well as the post-processing needed to obtain and decide on the optimal values are discussed in the subsections below.

2.3.4 Matlab Results and Post-Processing

Either definition of the problem yields, as stated before, the same results. We will, however, focus on the formulation that allows us to get all variables as a single LP. We start by computing the values for Pr_{Xi}, Pr_{Yi} and B_i. These values, for the parameters $E_1 = 5$, $E_2 = 14$ and $E_3 = 5$, have been computed in `MATLAB` using equations (12), (13) and (14), and are aggregated below:

i	Pr_{Xi} (€)	Pr_{Yi} (€)	B_i(€)
1	3,214	1,555	30,000
2	3,442	1,615	34,500
3	3,684	1,730	39,000
4	4,040	1,900	43,500
5	4,410	2,075	48,000
6	4,880	2,300	52,500
7	5,450	2,575	57,000
8	6,120	2,900	61,500
9	6,890	3,275	66,000
10	7,760	3,700	70,500

Table 2: Calculated total values of Pr_{Xi}, Pr_{Yi} and B_i for each of the 10 years.

With these values, we can construct matrices A and b as outlined in (23). Similarly, and knowing $P_x = 4000W$ and $P_Y = 2500W$, we can construct matrix c as stated in (22). One can now solve the LP problem using the following `MATLAB` command:

```
[x1c,~] = linprog(single_c,single_A,single_b,[],[]);
```
(24)

`x1c` is a vector which contains all 20 optimization variables. The output from the `linprog` command is organized below, rounded to 3 decimal places.

i	1	2	3	4	5	6	7	8	9	10
X_i	6.835	8.275	9.334	9.672	9.892	9.651	9.078	8.291	7.385	6.428
Y_i	5.164	3.724	2.665	2.327	2.107	2.348	2.921	3.708	4.614	5.571

Table 3: Optimization results for all 10 years, in units of X_i, Y_i

These are the optimal values for each year. However, as one can see, these values aren't feasible in real life as we can only have integer values of air conditioning units. These values must now be analysed in order to find the integer equivalents for each year, while making sure that we still meet the desired constraints.

To solve this problem, the values of X_i and Y_i were rounded up and down to obtain four possible pairs of values (X_i, Y_i) for each year. All possible pairs were tested, and those that didn't break any of the constraints were plugged in the optimization equation $f(X_i, Y_i)$. The pair of values that maximizes the power was then chosen as the number of units of X and Y for that particular year. This process is reproduced below for the first year.

$$(X_1, Y_1) = (6.835, 5.164) \Rightarrow \begin{cases} X = [6, 7] \\ Y = [5, 6] \end{cases} \Rightarrow \begin{cases} (X, Y)_1 = (6, 5) \\ (X, Y)_2 = (6, 6) \\ (X, Y)_3 = (7, 5) \\ (X, Y)_4 = (7, 6) \end{cases}$$

$$(X, Y)_3 \longrightarrow XPr_{Xi} + YPr_{Yi} = 30273 > 30000 \longrightarrow \times$$

$$(X, Y)_4 \longrightarrow X + Y = 13 > 12 \longrightarrow \times$$

$$(X, Y)_1 \longrightarrow XP_X + YP_Y = 36.5kW \longrightarrow \times$$

$$(X, Y)_2 \longrightarrow XP_X + YP_Y = 39.0kW \longrightarrow \checkmark$$

For year one, we'd then have $(X_1, Y_1) = (6, 6)$. The same processes was repeated for every year. The best possible pairs (X_i, Y_i), as well as the total power and the leftover budget are organized in the table below:

i	X_i	Y_i	P_{total} (kW)	ΔB_i (€)
1	6	6	39	1386
2	8	4	42	504
3	9	3	43.5	654
4	9	3	43.5	1440
5	9	3	43.5	2085
6	9	3	43.5	1680
7	9	3	43.5	225
8	8	4	42	940
9	7	5	40.5	1395
10	6	6	39	1740

Table 4: Results after post-processing of data.

2.3.5 Results and Discussion

From the point of view of merely optimizing power output, years 3 trough 7 all have the same solutions for values of X and Y units, $X = 9$ and $Y = 3$, for a total power output $P = 43.5$kW.

However, it is logical to switch as late as possible, because switching every 3 years would mean that the facility would have to spend $X_3 Pr_{X3} + Y_3 Pr_{Y3} = 38,346€$ every three years, or about $12,822€$ per year. A similar argument can be made to exclude keeping the air conditioners for 4, 5 or 6 years. When choosing the maximum number of years, 7, the facility gets the same amount of total power, while spending $56,775€$ every 7 years, the equivalent of about $8,111€$ per year.

Thus, in conclusion, the logical move that maximizes power output while being the most economically sensible for the facility would be to buy 9 units of air conditioner type X and 3 units of air conditioner type Y, and then keep them for a total of 7 years.

3 Task 2

Taking into consideration the differential equation consisting of the temperature inside the building, the dynamic model can be represented as:

$$\frac{dT_b(t)}{dt} = a_1\dot{q}_{solar}(t) + a_2[\dot{q}_{occ}(t) + \dot{q}_{ac}(t) - \dot{q}_{vent}(t)] + 3a_3[T_{amb}(t) - T_b(t)] \tag{25}$$

We discretize the above given equation to process the data that is available at discrete time instants ($t_k = k\Delta t$ where Δt is the sample interval):

$$\frac{dT_b(t)}{dt} \approx \frac{T_b(k+1) - T_{b,k}}{\Delta t}$$

Applying the above approximation, we translate (25) as follows:

$$\frac{T_b(k+1) - T_{b,k}}{\Delta t} \approx a_1\dot{q}_{solar}(t) + a_2[\dot{q}_{occ}(t) + \dot{q}_{ac}(t) - \dot{q}_{vent}(t)] + a_3[T_{amb}(t) - T_b(t)]$$

Further separating the respective coefficients in order to obtain the desired structure for the solution:

$$T_b(k+1) \approx (1 - a_3\Delta t)T_b(k) + \Delta t \begin{bmatrix} a_1 & a_2 & a_2 & -a_2 & a_3 \end{bmatrix} \begin{bmatrix} \dot{q}_{solar}(k) \\ \dot{q}_{occ}(k) \\ \dot{q}_{ac}(k) \\ \dot{q}_{vent}(k) \\ T_{amb}(k) \end{bmatrix} \tag{26}$$

We finally get the respective values for A and B, where A is a scalar, and B is a row vector, $B \in \mathbb{R}^{(1\times5)}$:

$$A = (1 - a_3\Delta t) \qquad B = \Delta t \begin{bmatrix} a_1 & a_2 & a_2 & -a_2 & a_3 \end{bmatrix} \tag{27}$$

4 Task 3

4.1 Problem Definition

We are given the task of:

$$\min_{a_1,a_2,a_3} \sum_{k=1}^{2159} \left(T_b(k+1) - \left(AT_b(k) + B\begin{bmatrix}\dot{q}_{solar}(k) & \dot{q}_{occ}(k) & \dot{q}_{ac}(k) & \dot{q}_{vent}(k) & T_{amb}(k)\end{bmatrix}^\intercal\right)\right)^2 \tag{28}$$

One can start by noticing this is an mean square error minimization problem. In fact, we can write (26) as:

$$T_b(k+1) = AT_b(k) + B\begin{bmatrix}\dot{q}_{solar}(k) & \dot{q}_{occ}(k) & \dot{q}_{ac}(k) & \dot{q}_{vent}(k) & T_{amb}(k)\end{bmatrix}^\intercal + e(k) \tag{29}$$

$$\Leftrightarrow$$

$$e(k) = T_b(k+1) - \left(AT_b(k) + B\begin{bmatrix}\dot{q}_{solar}(k) & \dot{q}_{occ}(k) & \dot{q}_{ac}(k) & \dot{q}_{vent}(k) & T_{amb}(k)\end{bmatrix}^\intercal\right) \tag{30}$$

What we are trying to do is find the best parameters, $\{a_1, a_2, a_3\}$ that minimize e_k in a square sense. If we take into account (27), we can rewrite the minimization problem (28) with respect to our minimization parameters:

$$\min_{a_1,a_2,a_3} \sum_{k=1}^{2159} \left((T_b(k+1) - T_b(k)) - \Delta t\begin{bmatrix}\dot{q}_{solar}(k) & \dot{q}_{occ}(k) + \dot{q}_{ac}(k) - \dot{q}_{vent}(k) & T_{amb}(k) - T_b(k)\end{bmatrix}\begin{bmatrix}a_1\\a_2\\a_3\end{bmatrix}\right)^2$$

4.2 Quadratic Programming

This problem can further be viewed as follows:

$$\min_{a_1,a_2,a_3} \sum_{k=1}^{2159} \hat{e}^2 = \min_{a_1,a_2,a_3} E^\intercal E$$

Where E is a $N \times 1$ matrix ($N = 2159$), with each individual sum of our minimization problem being a different column of E. This matrix takes the following form:

$$E = Y - \phi x$$

$$Y = \begin{bmatrix}T_b(2) - T_b(1)\\(\dots)\\T_b(N+1) - T_b(N)\end{bmatrix}; \phi = \Delta t\begin{bmatrix}\dot{q}_{solar}(1) & \dot{q}_{occ}(1) + \dot{q}_{ac}(1) - \dot{q}_{vent}(1) & T_{amb}(1) - T_b(1)\\(\dots) & (\dots) & (\dots)\\\dot{q}_{solar}(N) & \dot{q}_{occ}(N) + \dot{q}_{ac}(N) - \dot{q}_{vent}(N) & T_{amb}(N) - T_b(N)\end{bmatrix}; x = \begin{bmatrix}a_1\\a_2\\a_3\end{bmatrix}$$

Plugging this back into our minimization problem, we obtain:

$$\begin{aligned}
&\min_{a_1,a_2,a_3} E^\intercal E \Leftrightarrow \min_x (Y-\phi x)^\intercal (Y-\phi x) \Leftrightarrow \\
&\min_x Y^\intercal Y + x^\intercal \phi^\intercal \phi x - x^\intercal \phi^\intercal Y - Y^\intercal \phi x \Leftrightarrow \\
&\min_x Y^\intercal Y + \frac{1}{2} x^\intercal 2\phi^\intercal \phi x - 2Y^\intercal \phi x \Leftrightarrow \\
&\min_x d + \frac{1}{2} x^\intercal H x + c^\intercal x
\end{aligned} \tag{31}$$

Note: Because $x \in \mathbb{R}^{1\times 3}$, $\phi \in \mathbb{R}^{3\times K}$, $Y \in \mathbb{R}^{1\times K}$, the product $(x^\intercal \phi^\intercal Y)$ yields a scalar expression (denoted as N below). Therefore, the following applies:

$$x^\intercal \phi^\intercal Y = N \Leftrightarrow (x^\intercal \phi^\intercal Y)^\intercal = N^\intercal \Leftrightarrow Y^\intercal \phi x = N \Longrightarrow x^\intercal \phi^\intercal Y = Y^\intercal \phi x$$

In this form, it's easy to see this as an unconstrained optimization problem Where:

$$d = Y^\intercal Y = \begin{bmatrix} T_b(2) - T_b(1) \\ (\dots) \\ T_b(N+1) - T_b(N) \end{bmatrix}^\intercal \begin{bmatrix} T_b(2) - T_b(1) \\ (\dots) \\ T_b(N+1) - T_b(N) \end{bmatrix} \tag{32}$$

$$c = -2Y^\intercal \phi = \begin{bmatrix} T_b(2) - T_b(1) \\ (\dots) \\ T_b(N+1) - T_b(N) \end{bmatrix}^\intercal \begin{bmatrix} \dot{q}_{solar}(1) & \dot{q}_{occ}(1) + \dot{q}_{ac}(1) - \dot{q}_{vent}(1) & T_{amb}(1) - T_b(1) \\ (\dots) & (\dots) & (\dots) \\ \dot{q}_{solar}(N) & \dot{q}_{occ}(N) + \dot{q}_{ac}(N) - \dot{q}_{vent}(N) & T_{amb}(k) - T_b(N) \end{bmatrix} \tag{33}$$

$$H = 2\phi^\intercal \phi \qquad\qquad x = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \tag{34}$$

Further, since d is constant, it plays no role in the minimization problem, merely shifting the solution up or down. Therefore, we obtain, finally, the Unconstrained Quadratic Programming Minimization problem:

$$\min_x \frac{1}{2} x^\intercal H x + c^\intercal x \Leftrightarrow \min_{a_1,a_2,a_3} \frac{1}{2} \begin{bmatrix} a1 & a2 & a3 \end{bmatrix} 2\phi^\intercal \phi \begin{bmatrix} a1 \\ a2 \\ a3 \end{bmatrix} - 2Y^\intercal \phi \begin{bmatrix} a1 \\ a2 \\ a3 \end{bmatrix} \tag{35}$$

4.3 Results and Discussion

Having identified the problem as a Quadratic Programming and constructed the given H and c matrices, we now go ahead minimizing the problem to obtain the optimal values for the parameters

$\{a_1, a_2, a_3\}$. One can now solve the problem using the following `MATLAB` command:

$$\texttt{[x3,fval3,exitflag3] = quadprog(H3,c3)} \tag{36}$$

All parameters used for these calculations were taking specifically from the `measuremnts.csv` file. This gives rise to values of $\{a_1 = 4.56E-06,\ a_2 = -2.81E-05\ \ and\ \ a_3 = -4.12E-05\}$

The obtained optimal values of $\{a_1, a_2, a_3\}$ are used to calculate the A and B matrices as mentioned in (29) to obtain the MSE (Mean Square Error) of the function.

As a final remark, we notice that the minimization result is:

$$\sum_{k=1}^{2159}\left(T_b(k+1) - \left(AT_b(k) + B\left[\dot{q}_{solar}(k) \quad \dot{q}_{occ}(k) \quad \dot{q}_{ac}(k) \quad \dot{q}_{vent}(k) \quad T_{amb}(k)\right]^{\mathsf{T}}\right)\right)^2 = 3.9490E{+}04^{\circ}\mathrm{C}^2 \tag{37}$$

This value might seem very large when taken at face value. Therefore, we tried to inspect the results more closely. If we compute the values of $T_b(k+1)$ using the discovered values for $\{a_1, a_2, a_3\}$ and the values from `measurements.csv`, obtain a prediction of $T_b(k+1)$, which we can compare against the actual values. The graph below illustrates our results.

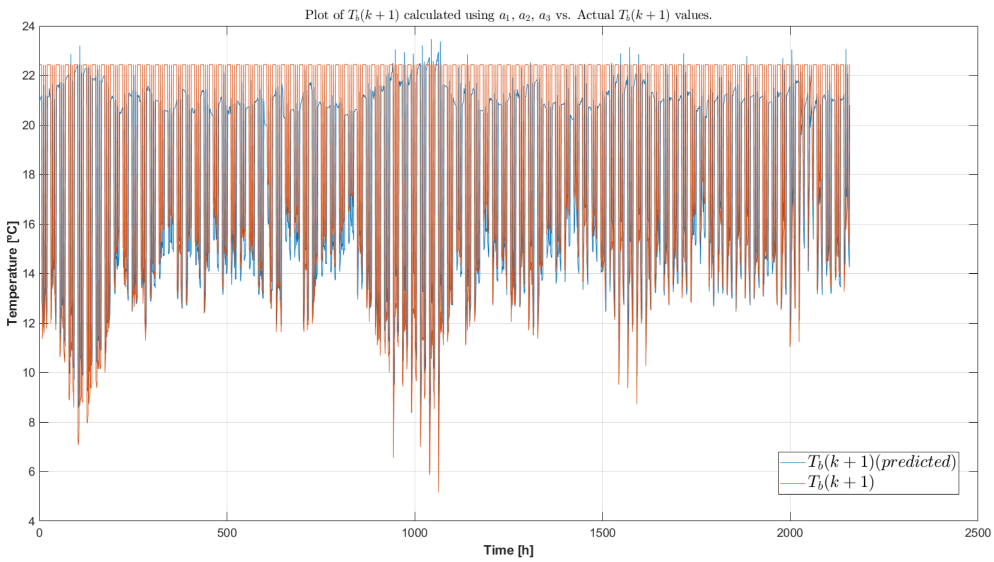

Figure 1: Plot of $T_b(k+1)$ calculated using a_1, a_2, a_3 vs. Actual $T_b(k+1)$ values.

We can see from this plot that, even though the lower predicted temperatures seem to be in line with the real ones, the constant upper temperatures have very noticeable variation to them comparatively to the real values. This makes sense, as for these hours, where the temperature stays essentially constant, we would need to have different values (a_1, a_2, a_3) in order to obtain the same results as the actual values of $T_b(k+1)$. We can conclude that, to have a perfect fit of $T_b(k+1)$, we'd need to vary (a_1, a_2, a_3) according to our hour, N. Since this isn't the case, the resulting fit and corresponding mean square error are justifiable.

5 Task 4

5.1 Problem Definiton

The minimization problem is presented as:

$$\min_{\substack{T_b(2),(\dots),T_b(N),\\ \dot{q}_{ac}(1),(\dots),\dot{q}_{ac}(N)}} \sum_{k=1}^{N} \left(\Phi(k)\dot{q}_{ac}(k)\Delta t + \frac{E_2+1}{10}(T_b(k) - T_{ref})^2 \right) \tag{38}$$

$$f(\dot{q}_{ac}, T_b) = \sum_{k=1}^{N} \left(\Phi(k)\dot{q}_{ac}(k)\Delta t + \frac{E_2+1}{10}(T_b(k) - T_{ref})^2 \right) \tag{39}$$

Over a time horizon of N steps, the task is to optimize the values of $\dot{q}_{ac}$ and T_b in order to minimize the cost of air conditioning for the building. Additionally, it is asked that, for every step, the dynamic model of the system must be kept, and the minimum and maximum values for $\dot{q}_{ac}$ and T_b must also be respected. Specifically for the values of T_b, the limits only need to be applied when there are people in the facility, that is, when $\dot{q}_{occ} > 0$. The constraints, therefore, are as follows:

$$T_b(k+1) = AT_b(k) + B \begin{bmatrix} \dot{q}_{solar}(k) & \dot{q}_{occ}(k) & \dot{q}_{ac}(k) & \dot{q}_{vent}(k) & T_{amb}(k) \end{bmatrix}^{\mathsf{T}} \qquad k = 1, ..., N-1 \tag{40}$$

$$0 \leq \dot{q}_{ac}(k) \leq \dot{q}_{ac,max} \qquad \dot{q}_{ac,max} = 100\text{kW} \tag{41}$$

$$T_{min} \leq T_b(k) \leq T_{min} \ \text{ when } \ \dot{q}_{occ} > 0, \quad \begin{cases} T_{min} = 15^{\circ}\text{C} \\ T_{max} = 28^{\circ}\text{C} \end{cases} \tag{42}$$

Finally, we are given values for all parameters not being optimized here. $T_b(1) = 22.43^{\circ}$C and $T_{ref} = 22^{\circ}$C. All other values can be retrieved from the `measurements.csv` file.

5.2 Quadratic Programming

We start by acknowledging that the given function, $f(\dot{q}_{ac}, T_b)$, is quadratic in nature, as the terms in T_b are subject to a square term. Looking at the given constraints we can identify our equality constraints as affine, and our inequality constraints as linear. We are, therefore, looking at a quadratic problem with linear/affine constraints, which is, indeed, a Quadratic Programming Problem. The rest of this section will focus on re-writing the problem as a standard quadratic problem.

5.2.1 Minimization Function

For compactness of notation, the following variable is introduced:

$$C = \frac{E_2 + 1}{10}$$

The minimization function, $f(\dot{q}_{ac}, T_b)$ can be re-worked as follows:

$$f(\dot{q}_{ac}, T_b) = \sum_{k=1}^{N} (\Phi(k)\dot{q}_{ac}(k)\Delta t) + \sum_{k=1}^{N} \left(C(T_b(k) - T_{ref})^2\right)$$
$$f(\dot{q}_{ac}, T_b) = \sum_{k=1}^{N} (\Phi(k)\dot{q}_{ac}(k)\Delta t) + \sum_{k=2}^{N} \left(C(T_b(k) - T_{ref})^2\right) + C(T_b(1) - T_{ref})^2$$
$$f(\dot{q}_{ac}, T_b) = \sum_{k=1}^{N} (\Phi(k)\dot{q}_{ac}(k)\Delta t) + \sum_{k=2}^{N} \left(C(T_b(k)^2 - 2T_{ref}T_b(k))\right) + C(\sum_{k=2}^{N}(T_{ref})^2 + (T_b(1) - T_{ref})^2)$$

One can notice that, by re-writing our function like this, one of the parcels is made up constants, which don't change the solution of the optimization problem at hand and can, therefore, be dropped. Further, we can now define our variables vector, x, as:

$$x = [T_b(2), (\dots), T_b(N), \dot{q}_{ac}(1), (\dots), \dot{q}_{ac}(N)]^\intercal, \qquad x \in \mathbb{R}^{(2N-1)\times 1} \tag{43}$$

The minimization function can then be written as:

$$f(x) = \sum_{k=1}^{N} (\Phi(k)x(N + k - 1)\Delta t) + \sum_{k=2}^{N} \left(C(x(k)^2 - 2T_{ref}x(k)\right)$$

We can now separate the linear and quadratic terms and re-write the function in vectorial form:

$$f(x) = \Big[\underbrace{-2CT_{ref} \quad (\cdots) \quad -2CT_{ref}}_{N-1} \quad \Phi(1)\Delta_t \quad (\cdots) \quad \Phi(N)\Delta_t\Big] x + x^\intercal C \left[\begin{array}{c|c} I_{N-1} & 0_{N-1,N} \\ \hline 0_{N,N-1} & 0_N \end{array}\right] x$$

Now, one can define the c and H matrices such that the problem takes the following form:

$$\min_x \frac{1}{2} x^\intercal H x + c^\intercal x$$

$$H = 2C \left[\begin{array}{c|c} (I_{N-1}) & 0_{N-1,N} \\ \hline 0_{N,N-1} & 0_N \end{array}\right] \tag{44}$$

$$c^\intercal = \Big[\underbrace{-2CT_{ref} \quad (\cdots) \quad -2CT_{ref}}_{N-1} \quad \Phi(1)\Delta_t \quad (\cdots) \quad \Phi(N)\Delta_t\Big] \tag{45}$$

5.2.2 Equality Constraints

When implementing the equality constraints, one starts by re-writing (40) with respect to our variables vector, x, for $k = 1, ..., N-1$:

$$x(k+1) = (1-a_3\Delta t)x(k) + \Delta t(a_1\dot{q}_{solar}(k) + a_2\dot{q}_{occ}(k) + a_2x(N+k-1) - a_2\dot{q}_{vent}(k) + a_3T_{amb}(k)) \tag{46}$$

Isolating the variables present in x, the $N-1$ equality equations can be re-written to take the matrix form $A_{eq}x = b_{eq}$, with $A_{eq} \in \mathbb{R}^{(N-1)\times(2N-1)}$ and $b_{eq} \in \mathbb{R}^{1\times(N-1)}$:

$$A_{eq} = \begin{bmatrix} 1 & 0 & (\cdots) & 0 & 0 & -a_2\Delta t & 0 & (\cdots) & 0 \\ a_3\Delta t - 1 & 1 & (\cdots) & 0 & 0 & 0 & -a_2\Delta t & (\cdots) & 0 \\ (\cdots) & (\cdots) & (\cdots) & (\cdots) & (\cdots) & (\cdots) & (\cdots) & (\cdots) & (\cdots) \\ 0 & 0 & (\cdots) & a_3\Delta t - 1 & 1 & 0 & 0 & (\cdots) & -a_2\Delta t \end{bmatrix} \tag{47}$$

$$b_{eq} = \begin{bmatrix} (1-a_3\Delta t)T_b(1) + \Delta t(a_1\dot{q}_{solar}(1) + a_2(\dot{q}_{occ}(1) - \dot{q}_{vent}(1)) + a_3T_{amb}(1)) \\ \Delta t(a_1\dot{q}_{solar}(2) + a_2(\dot{q}_{occ}(2) - \dot{q}_{vent}(2)) + a_3T_{amb}(2)) \\ (\cdots) \\ \Delta t(a_1\dot{q}_{solar}(N-1) + a_2(\dot{q}_{occ}(N-1) - \dot{q}_{vent}(N-1)) + a_3T_{amb}(N-1)) \end{bmatrix} \tag{48}$$

5.2.3 Inequality Constraints

Similar to what was done in the last subsection, we can write (41) and (42) as the four following inequations:

$$\begin{cases} x(k) \le T_{max}, & k = 1, 2, (\dots), N-1 \\ -x(k) \le -T_{min}, & k = 1, 2, (\dots), N-1 \\ x(k) \le \dot{q}_{ac,max}, & k = N, N+1, (\dots)2N-1 \\ -x(k) \le 0, & k = N, N+1, (\dots), 2N-1 \end{cases} \tag{49}$$

Writing the inequalities above in matrix form, we get the set $A_{ineq}x \le b_{ineq}$, where $A_{ineq} \in \mathbb{R}^{(N-1)\times(4N-2)}$ and $b_{ineq} \in \mathbb{R}^{1\times(4N-2)}$:

$$A_{ineq} = \begin{bmatrix} I_{(N-1,)} & 0_{(N-1,N)} \\ -I_{N-1} & 0_{(N-1,N)} \\ 0_{(N,N-1)} & I_N \\ 0_{(N,N-1)} & -I_N \end{bmatrix} \qquad b_{ineq} = \begin{bmatrix} T_{max} \\ -T_{min} \\ \dot{q}_{ac,max} \\ 0_{(N\times 1)} \end{bmatrix} \tag{50}$$

Further, for entries in which the value of $\dot{q}_{occ}(N) = 0$, the corresponding temperature rows in A_{ineq} and entries in b_{ineq} should be removed, because this condition does not apply at these points.

With H (44), c (45), A_{eq} (47), b_{eq} (48), A_{ineq} b_{ineq} (50), our quadratic programming problem is now fully defined and constrained, and can be solved using the `quadprog` command in `MATLAB`.

5.3 Alternative Formulations of the Quadratic Programming Problem

By re-writing the given constraints equation in a way to evidence $\dot{q}_{ac}$, we can replace instances of $\dot{q}_{ac}(k)$ by instances of $T_b(k)$ and $T_b(k+1)$ for $k = 1, 2, ...N-1$, considerably reducing the number of variables in our optimization vector, x. Further, since the constraints equation gets incorporated into the minimization target $f(x)$, we can eliminate nearly all equality constraints from our problem. This formulation of the problem, though potentially much faster to compute, due to the reduced constraints and variables, ended up causing some problems computationally, taking substantially longer to solve in `MATLAB`, using `quadprog`, than the formulation presented in the section above. Nevertheless, it did yield the exact same results, and is thus briefly presented below:

5.3.1 Minimization Function

Writing the equality constraint as:

$$\dot{q}_{ac}(k) = \frac{T_b(k+1) + (a_3 - 1\Delta t)T_b(k) - \Delta t(a_1\dot{q}_{solar}(k) + a_2(\dot{q}_{occ}(k) - \dot{q}_{vent}(k)) + a_3T_{amb}(k))}{a_2\Delta t} \tag{51}$$

We can replace $\dot{q}_{ac}(k)$ in $f(\dot{q}_{ac}, T_b)$. Doing that and dropping all constant values, which don't affect our problem, results in the following formulation:

$$f(\dot{q}_{ac}, T_b) = \sum_{k=1}^{N-1} \left(\frac{\Phi(k)}{a_2}(T_b(k+1) + (a_3 - 1)T_b(k)) \right) + \sum_{k=1}^{N} \left(CT_b(k)^2 - 2CT_{ref}T_B(k) \right) + \Phi(N)\Delta_t\dot{q}_{ac}(N)$$

Defining the optimization vector, $x \in \mathbb{R}^{N\times1}$ as:

$$x = \begin{bmatrix} T_b(2) & T_b(3) & (\cdots) & T_b(N) & \dot{q}_{ac}(N) \end{bmatrix} \tag{52}$$

One can obtain the following H and c matrices for the Quadratic Minimization Problem, where

$H \in \mathbb{R}^{N\times N}$ and $c \in \mathbb{R}^{N\times 1}$:

$$H = 2C\left[\begin{array}{ccc|c} \ddots & & & | \\ & I_{(N-1)} & & 0_{(N-1)\times 1} \\ & & \ddots & | \\ \hline —— & 0_{1\times(N-1)} & —— & 0 \end{array}\right],\ c = \begin{bmatrix} \frac{\Phi(1)}{a_2} - (\frac{A\Phi(2)}{a_2} + 2CT_{ref}) \\ \frac{\Phi(2)}{a_2} - (\frac{A\Phi(3)}{a_2} + 2CT_{ref}) \\ (\cdots) \\ \frac{\Phi(N-2)}{a_2} - (\frac{A\Phi(N-1)}{a_2} + 2CT_{ref}) \\ \frac{\Phi(N-1)}{a_2} - 2CT_{ref} \\ \Delta t\Phi(N) \end{bmatrix} \tag{53}$$

5.3.2 Equality and Inequality Constraints

All of our equality constraints have been eliminated, however, the inequality constraints are still present. We can write (49) as:

$$\begin{cases} T_b(k) \leq T_{max} \\ -T_b(k) \leq -T_{min} \\ \dot{q}_{ac} \leq \dot{q}_{ac,max} \\ -\dot{q}_{ac} \leq 0 \end{cases} \Leftrightarrow \begin{cases} T_b(k) \leq T_{max} \\ -T_b(k) \leq -T_{min} \\ \frac{T_b(k+1)-AT_b(k)}{a_2\Delta t} \leq \dot{q}_{ac,max} + \dot{q}_{occ}(k) - \dot{q}_{vent}(k) - \frac{a_1}{a_2}\dot{q}_{solar}(k) + \frac{a_3}{a_2}T_{amb}(k) \\ \frac{AT_b(k)-T_b(k+1)}{a_2\Delta t} \leq \dot{q}_{occ}(k) - \dot{q}_{vent}(k) - \frac{a_1}{a_2}\dot{q}_{solar}(k) + \frac{a_3}{a_2}T_{amb}(k) \\ 0 \leq \dot{q}_{ac}(N) \leq \dot{q}_{ac,max} \end{cases}$$

This formulation allows us to build the matrices $A_{ineq} \in \mathbb{R}^{(4N+2)\times N}$ and $b_{ineq} \in \mathbb{R}^{1\times(4N+2)}$:

$$A_{ineq} = \begin{bmatrix} 1 & 0 & (\cdots) & 0 & 0 & 0 \\ (\cdots) & (\cdots) & (\cdots) & (\cdots) & (\cdots) & (\cdots) \\ 0 & 0 & (\cdots) & 0 & 1 & 0 \\ -1 & 0 & (\cdots) & 0 & 0 & 0 \\ (\cdots) & (\cdots) & (\cdots) & (\cdots) & (\cdots) & (\cdots) \\ 0 & 0 & (\cdots) & 0 & -1 & 0 \\ \frac{1}{\Delta t a_2} & 0 & (\cdots) & 0 & 0 & 0 \\ -\frac{A}{\Delta t a_2} & \frac{1}{\Delta t a_2} & (\cdots) & 0 & 0 & 0 \\ (\cdots) & (\cdots) & (\cdots) & (\cdots) & (\cdots) & (\cdots) \\ 0 & 0 & (\cdots) & -\frac{A}{\Delta t a_2} & \frac{1}{\Delta t a_2} & 0 \\ -\frac{1}{\Delta t a_2} & 0 & (\cdots) & 0 & 0 & 0 \\ \frac{A}{\Delta t a_2} & -\frac{1}{\Delta t a_2} & (\cdots) & 0 & 0 & 0 \\ (\cdots) & (\cdots) & (\cdots) & (\cdots) & (\cdots) & (\cdots) \\ 0 & 0 & (\cdots) & \frac{A}{\Delta t a_2} & -\frac{1}{\Delta t a_2} & 0 \\ 0 & 0 & (\cdots) & 0 & 0 & 1 \\ 0 & 0 & (\cdots) & 0 & 0 & -1 \end{bmatrix} \tag{54}$$

$$b_{ineq} = \begin{bmatrix} T_{max} \\ (\cdots) \\ T_{max} \\ -T_{min} \\ (\cdots) \\ -T_{min} \\ \frac{AT_b(1)}{a_2\Delta t} + \dot{q}_{ac,max} + \dot{q}_{occ}(1) - \dot{q}_{vent}(1) - \frac{a_1}{a_2}\dot{q}_{solar}(1) + \frac{a_3}{a_2}T_{amb}(1) \\ \dot{q}_{ac,max} + \dot{q}_{occ}(2) - \dot{q}_{vent}(2) - \frac{a_1}{a_2}\dot{q}_{solar}(2) + \frac{a_3}{a_2}T_{amb}(2) \\ (\cdots) \\ \dot{q}_{ac,max} + \dot{q}_{occ}(N) - \dot{q}_{vent}(N) - \frac{a_1}{a_2}\dot{q}_{solar}(N) + \frac{a_3}{a_2}T_{amb}(N) \\ -\frac{AT_b(1)}{a_2\Delta t} + \dot{q}_{ac,max} + \dot{q}_{occ}(1) - \dot{q}_{vent}(1) - \frac{a_1}{a_2}\dot{q}_{solar}(1) + \frac{a_3}{a_2}T_{amb}(1) \\ \dot{q}_{ac,max} + \dot{q}_{occ}(2) - \dot{q}_{vent}(2) - \frac{a_1}{a_2}\dot{q}_{solar}(2) + \frac{a_3}{a_2}T_{amb}(2) \\ (\cdots) \\ \dot{q}_{ac,max} + \dot{q}_{occ}(N) - \dot{q}_{vent}(N) - \frac{a_1}{a_2}\dot{q}_{solar}(N) + \frac{a_3}{a_2}T_{amb}(N) \\ 100 \\ 0 \end{bmatrix} \tag{55}$$

The problem is now fully defined. As stated above, however, it would seem that, despite both methods yielding the same results, one would expect this alternative method to be faster than the one presented in the last section, because it has less than half of the optimization variables. However, it would seem that the added complexity of the inequality constraints actually slows down the solver enough to make this alternative solution slower than the method formulated in 5.2.

As a final note about this formulation, it was noticed that, if the constraints on the maximum and minimum of $\dot{q}_{ac}$ were ignored while using the values from `measurements.csv`, the solution obtained would be the exact same, but the solver would finish in a fraction of the time. This leads the authors of this report to believe that, for this specific set of measurements, those constraints would not be needed. However, this obviously would leave $\dot{q}_{ac}$ unconstrained, which could lead to values above or below the desired maximum and minimum for any other set of measurements.

5.4 Matlab Results and Post-Processing

5.4.1 Matlab Code Implementation

Both methods described above were implemented using `MATLAB`. As the alternative solution 5.3 took considerably longer than the one defined in 5.2, the corresponding code section of the former has been commented out, and the latter will be the focus of the remainder of this section. As for the Quadratic Programming Problem described in 5.2, it was solved using the command `quadprog` as follows:

```
[x4,~] = quadprog(H,c,[],[],Aeq,beq,lb,ub,[],options);
```
(56)

In the above instruction, `x4`, `H`, `c`, `Aeq` and `beq` are the computational implementations of x (43), H (44), c (45), A_{eq} (47) and b_{eq} (48), respectively.

All other necessary values were either taken from the given description of Task 4 in the provided PDF, or from the `measurements.csv` file. One should note that, for both convenience and for compatibility of units, the values given for Φ were divided by 3600, to convert these values form $\left[\frac{€}{\text{kWh}}\right]$ to $\left[\frac{€}{\text{kJ}}\right]$.

For ease of coding, and because the inequalities were merely upper and lower bounds on the values of x, the choice was made to implement these not as inequality matrices A_{ineq} and b_{ineq}, but as vectors `lb` and `ub`, that hold the values for the lower and upper bounds, respectively. These two vectors were defined as follows:

$$lb = \begin{bmatrix} T_{min} & (\cdots) & T_{min}, 0, (\cdots), 0 \end{bmatrix} \tag{57}$$

$$ub = \begin{bmatrix} T_{max} & (\cdots) & T_{max}, \dot{q}_{ac,max}, (\cdots), \dot{q}_{ac,max} \end{bmatrix} \tag{58}$$

Further, the first $N-1$ entries of both vectors, corresponding to the maximum and minimum values of $T_b(2)$ to $T_b(N)$, were checked against the $\dot{q}_{occ}$ values from the `measurements.csv` file. For all null entries of $\dot{q}_{occ}$, the corresponding upper and lower bounds were removed by replacing T_{max}/T_{min} values with $+\infty/-\infty$.

5.4.2 Results

After the problem was solved, the solution vector `x4` was split in two. The first $N-1$ entries were concatenated with the given value for $T_b(1)$ in the vector `T_b_opt`. The last N entries of `x4` were assigned to the vector `q_dot_ac_opt`. A vector called `hours` was created, spanning values $k = 1, ..., N$, and was used to graph both optimized vectors against. The results are plotted bellow:

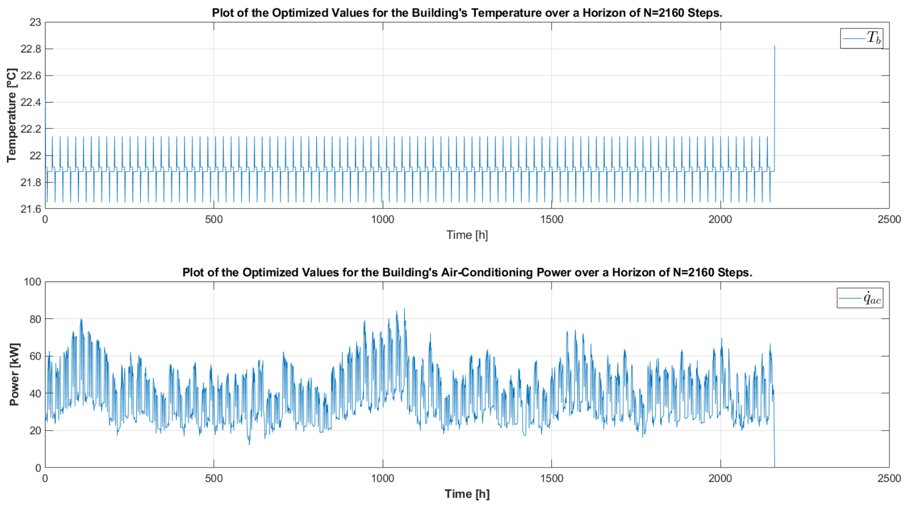

Figure 2: Plot of T_b and $\dot{q}_{ac}$ over the horizon of $N = 2160$ steps

With these values, the original cost function to minimize can now be calculated. Below is plot of $f(T_b, \dot{q}_{ac})$, the cost of air-conditioning, for every hour for the 90 days of our problem.

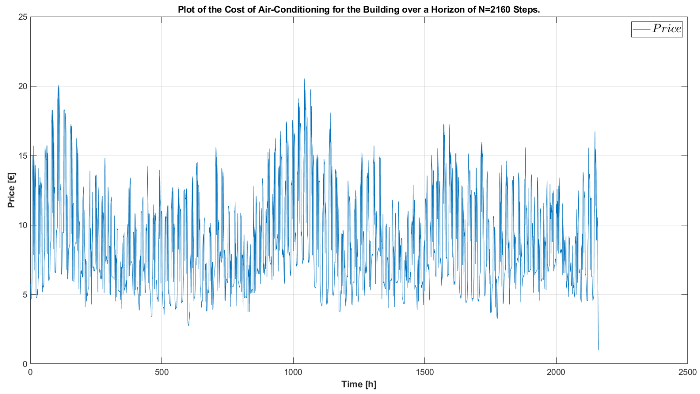

Figure 3: Plot of $f(T_b, \dot{q}_{ac})$ over the horizon of $N = 2160$ steps

Perhaps more importantly, the result of the minimization function, representing the cost of air-conditioning over the 90 days is:

$$f(\dot{q}_{ac}, T_b) = \sum_{k=1}^{N} \left(\Phi(k)\dot{q}_{ac}(k)\Delta t + C(T_b(k) - T_{ref})^2\right) = 19,227.29 \text{ €} \tag{59}$$

5.4.3 Discussion

Let us start this discussion section by replacing the values of $E2$ and Φ by they're corresponding numerical values (in the case of Φ, we'll be replacing it for the maximum possible price of air conditioning, taken from the `measurements.csv` file):

$$f(\dot{q}_{ac}, T_b) = \sum_{k=1}^{N} \left(0.25\dot{q}_{ac}(k) + 3(T_b(k) - T_{ref})^2\right)$$

Looking at the equation, we can see that values of $\dot{q}_{ac}$ are priced, at most, at a factor of 0.25 per kW. On the other hand, any deviation from of T_b form T_{ref} is not only penalized by a factor of 3, but whatever that deviation is also gets squared, increasing the problem still. Further, if $\dot{q}_{ac}(k) = \dot{q}_{ac,max}$ for one time step, the cost of that is $0.25 \times 100 = 25$€. Comparatively, a change of $\Delta T_b = 3$°C results in a price of $3 \times (\Delta T_b)^2 = 27$€. We can easily see that it wouldn't be useful at all to deviate too much from T_{ref}. For instance, if one were to try and minimize the direct cost of air-conditioning and just keep the temperatures at minimum levels, one would actually be heavily penalised as, for every time step, we would be incurring in the indirect cost of $3 \times (T_{min} - T_{ref})^2 = 147$€.

Therefore, when trying to minimize $f(\dot{q}_{ac}, T_b)$, it makes sense to maintain T_b as close to T_{ref} as possible, in order to avoid that penalization, and vary $\dot{q}_{ac}$ in a way as to keep T_b close to 22°C.

In practise, results show exactly that. The maximum deviation from T_{ref} happens at points where $T_b = 21.65$°C, a mere 0.45°C of deviation. For most hours in our horizon, $T_b \in [21.88; 21.91]$°C, and the maximum temperature reached is $T_b = 22.14$°C. By contrast, values for $\dot{q}_{ac}$ vary wildly to keep temperatures at optimal values.

Further, plots of $\dot{q}_{ac}$ and T_{amb} appear to make physical sense. A falling trend in ambient temperature is accompanied by a rising trend in values of $\dot{q}_{ac}$. Looking at the micro level, a plot of $\dot{q}_{ac}$ and $\dot{q}_{occ}$ reveals that air conditioning power drops when people are not in the building and at night, which, again makes physical sense. Both of these observations can be checked in the figures below.

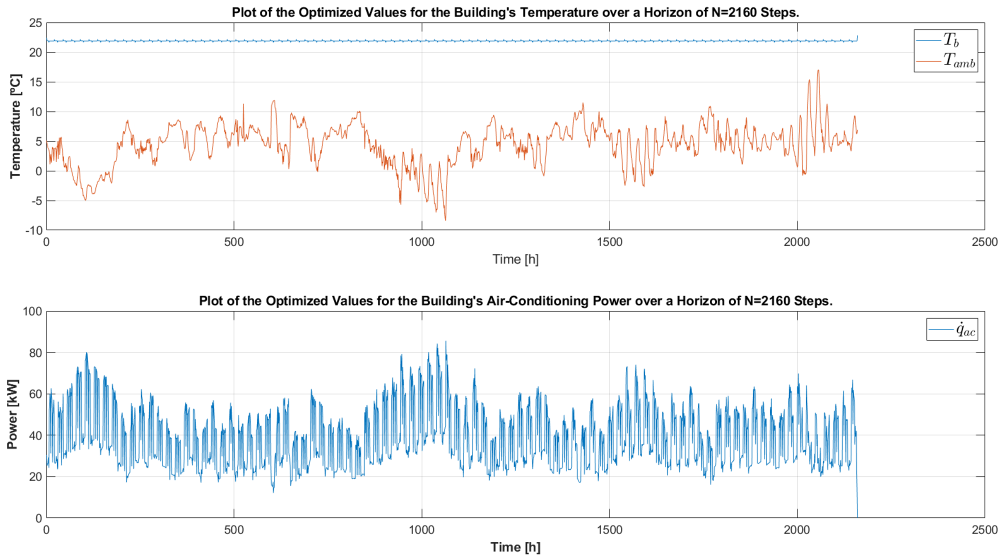

Figure 4: Plot of T_b, T_{amb} and $\dot{q}_{ac}$ over the horizon of $N = 2160$ steps.

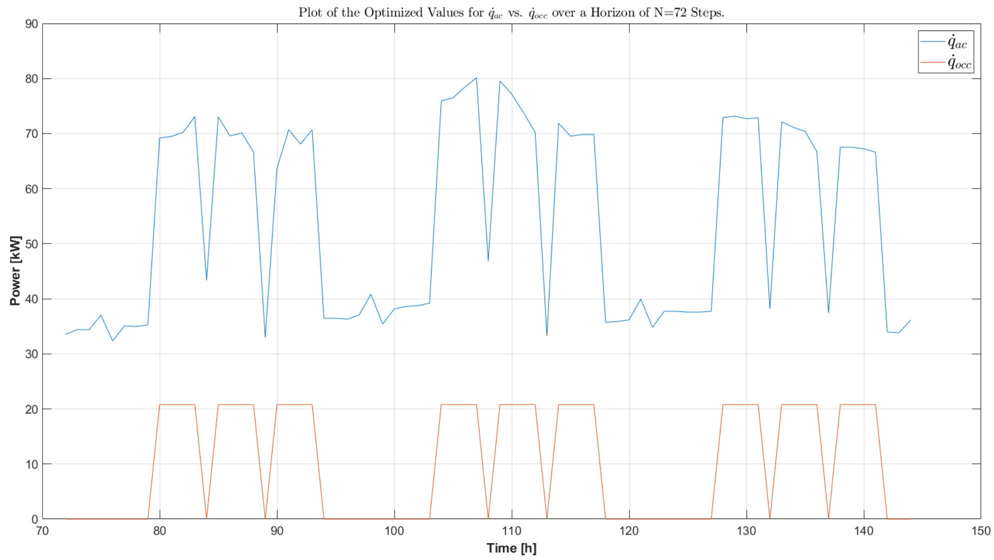

Figure 5: Plot of the Optimized Values for $\dot{q}_{ac}$ vs. $\dot{q}_{occ}$ over a Horizon of N=72 steps.

Finally, by inspection, we can see that the constraints were indeed kept. $\dot{q}_{occ}$ was kept between 0 and 100kW, T_b between 15^{o}c and 28^{o}c. Calculation of the equality constraints reveals the dynamic model was indeed kept for all time steps in the considered horizon.

The obtained results seem to make sense from both a mathematical and a physical stand-point and that all constraints were met, one can assume the function was successfully optimised and that the smallest possible value for the cost of air conditing over the 3 months is, indeed $19,227.29$€.

6 References

1. T. van den Boom and B. De Schutter, Optimization in Systems and Control. Delft Center for Systems and Control, Delft University of Technology, 2018.

2. B. De Schutter, Optimization for Systems and Control Lecture Slides [PDF File]. 2020-2021.